IEEE *Xplore* ℗
RELEASE 2.1

Welcome United States Patent and Trademark Office

⌐□⌐Search Session History

BROWSE   **SEARCH**   **IEEE XPLORE GUIDE**   **SUPPORT**

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Thu, 9 Mar 2006, 5:55:05 PM EST

Search Query Display

██████ ███

Recent Search Queries   **Results**

#1	((attribute cache <or> attribute caching)<in>metadata)	1
#2	((attribute cache)<in>metadata)	1
#3	((attribute cache <or> attribute caching)<in>metadata)	1
#4	(((attribute cache <or> attribute caching) <and> tlb) <in>metadata)	0

████████████

Help   Contact Us   Privacy & Security   IEEE.org

Indexed by
⚇ Inspec˙

http://ieeexplore.ieee.org/search/history.jsp3/9/06

💉 Feedback Report a problem Satisfaction survey

Terms used **attribute cache** or **attribute caching**   Found **42** of **171,143**

Sort results by	relevance 🔽	❤ Save results to a Binder	Try an Advanced Search
Display results	expanded form 🔽	⁇ Search Tips ☐ Open results in a new window	Try this search in The ACM Guide

Results 1 - 20 of 42   Result page: **1**   2   3   next

Relevance scale ☐⬜⬛⬛⬛

1   Event matching in symmetric subscription systems   ⬛
Walid Rjaibi, Klaus R. Dittrich, Dieter Jaepel
September 2002 **Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research**
Publisher: IBM Press

Full text available: 📄 pdf(192.61 KB)   Additional Information: full citation, abstract, references, citings, index terms

Publish/subscribe and database systems researchers have recognized the importance of the event matching algorithm to the performance and scalability of a content-based subscription system. A number of interesting event matching techniques as well as DBMS solutions have been proposed in recent research work in the area. Content-based subscription systems allow information consumers to define filtering criteria when they register their interest in being notified of events that match their requirem ...

2   A quantitative analysis of cache policies for scalable network file systems   ⬛
◈ Michael D. Dahlin, Clifford J. Mather, Randolph Y. Wang, Thomas E. Anderson, David A. Patterson
May 1994 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1994 ACM SIGMETRICS conference on Measurement-and modeling of computer systems SIGMETRICS '94**, Volume 22 Issue 1
Publisher: ACM Press

Full text available: 📄 pdf(1.42 MB)   Additional Information: full citation, abstract, references, citings, index terms

Current network file system protocols rely heavily on a central server to coordinate file activity among client workstations. This central server can become a bottleneck that limits scalability for environments with large numbers of clients. In central server systems such as NFS and AFS, all client writes, cache misses, and coherence messages are handled by the server. To keep up with this workload, expensive server machines are needed, configured with high-performance CPUs, memory systems, ...

3   Performance of cache coherence in stackable filing   ⬛
◈ J. Heidemann, G. Popek
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles SOSP '95**, Volume 29 Issue 5
Publisher: ACM Press
Full text available: 📄 pdf(2.00 MB)   Additional Information: full citation, references, index terms

4 On adaptive caching in mobile databases

Hong V. Leong, Antonio Si
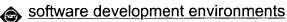
April 1997 **Proceedings of the 1997 ACM symposium on Applied computing**

Publisher: ACM Press

Full text available: pdf(873.16 KB)   Additional Information: full citation, references, citings, index terms

Keywords: cache replacement, data caching, mobile databases, query processing

---

5 Parallel and distributed incremental attribute evaluation algorithms for multiuser software development environments

Gail E. Kaiser, Simon M. Kaplan

January 1993 **ACM Transactions on Software Engineering and Methodology (TOSEM),** Volume 2 Issue 1

Publisher: ACM Press

Full text available: pdf(3.09 MB)   Additional Information: full citation, abstract, references, citings, index terms

The problem of change propagation in multiuser software development environments distributed across a local-area network is addressed. The program is modeled as an attributed parse tree segmented among multiple user processes and changes are modeled as subtree replacements requested asynchronously by individual users. Change propagation is then implemented using decentralized incremental evaluation of an attribute grammar that defines the static semantic properties of the p ...

Keywords: attribute grammar, change propagation, distributed, incremental algorithm, parallel, reliability

---

6 Extensible file systems in spring

Yousef A. Khalidi, Michael N. Nelson

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles SOSP '93,** Volume 27 Issue 5
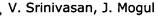
Publisher: ACM Press

Full text available: pdf(1.47 MB)   Additional Information: full citation, abstract, references, citings, index terms

In this paper we describe an architecture for extensible file systems. The architecture enables the extension of file system functionality by composing (or stacking) new file systems on top of existing file systems. A file system that is stacked on top of an existing file system can access the existing file system's files via a well-defined naming interface and can share the same underlying file data in a coherent manner. We describe extending file systems in the context of the Spring operating ...

---

7 Spritely NFS: experiments with cache-consistency protocols

V. Srinivasan, J. Mogul

November 1989 **ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles SOSP '89,** Volume 23 Issue 5

Publisher: ACM Press

Full text available: pdf(1.50 MB)   Additional Information: full citation, abstract, references, citings, index terms

File caching is essential to good performance in a distributed system, especially as processor speeds and memory sizes continue to improve rapidly while disk latencies do not. Stateless-server systems, such as NFS, cannot properly manage client file caches. Stateful systems, such as Sprite, can use explicit cache consistency protocols to improve both cache consistency and overall performance. By modifying NFS to use the Sprite cache

consistency protocols, we isolate the effects o ...

8 The ergo support system: an integrated set of tools for prototyping integrated ■
environments

Peter Lee, Frank Pfenning, Gene Rollins, William Scherlis

November 1988 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices ,
Proceedings of the third ACM SIGSOFT/SIGPLAN software
engineering symposium on Practical software development
environments SDE 3**, Volume 13 , 24 Issue 5 , 2

Publisher: ACM Press

Full text available: 📄 pdf(1.07 MB)   Additional Information: full citation, abstract, references, citings, index terms

The Ergo Support System (ESS) is an engineering framework for experimentation and
prototyping to support the application of formal methods to program development,
ranging from program analysis and derivation to proof-theoretic approaches. The ESS is a
growing suite of tools that are linked together by means of a set of abstract interfaces.
The principal engineering challenge is the design of abstract interfaces that are
semantically rich and yet flexible enough to permit experimentation wit ...

9 Separating key management from file system security ■

David Mazières, Michael Kaminsky, M. Frans Kaashoek, Emmett Witchel

December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the
seventeenth ACM symposium on Operating systems principles SOSP
'99**, Volume 33 Issue 5

Publisher: ACM Press

Full text available: 📄 pdf(1.77 MB)   Additional Information: full citation, abstract, references, citings, index terms

No secure network file system has ever grown to span the Internet. Existing systems all
lack adequate key management for security at a global scale. Given the diversity of the
Internet, any particular mechanism a file system employs to manage keys will fail to
support many types of use.We propose separating key management from file system
security, letting the world share a single global file system no matter how individuals
manage keys. We present SFS, a secure file system that avoids internal ...

10 The Zebra striped network file system ■

John H. Hartman, John K. Ousterhout

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the
fourteenth ACM symposium on Operating systems principles SOSP
'93**, Volume 27 Issue 5

Publisher: ACM Press

Full text available: 📄 pdf(1.93 MB)   Additional Information: full citation, abstract, references, citings, index terms

Zebra is a network file system that increases throughput by striping file data across
multiple servers. Rather than striping each file separately, Zebra forms all the new data
from each client into a single stream, which it then stripes using an approach similar to a
log-structured file system. This provides high performance for writes of small files as well
as for reads and writes of large files. Zebra also writes parity information in each stripe in
the style of RAID disk arrays; this increase ...

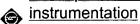**11** On implementing MPI-IO portably and with high performance ■

Rajeev Thakur, William Gropp, Ewing Lusk

May 1999 **Proceedings of the sixth workshop on I/O in parallel and distributed
systems**

Publisher: ACM Press

Full text available: 📄 pdf(887.89 KB)   Additional Information: full citation, references, citings, index terms

12 A comparison of system monitoring methods, passive network monitoring and kernel instrumentation

A. W. Moore, A. J. McGregor, J. W. Breen

January 1996 **ACM SIGOPS Operating Systems Review**, Volume 30 Issue 1

Publisher: ACM Press

Full text available: ▓ pdf(1.89 MB)   Additional Information: full citation, abstract, index terms

This paper presents the comparison of two methods of system monitoring, passive network monitoring and kernel instrumentation. The comparison is made on the basis of passive network monitoring being used as a replacement for kernel instrumentation in some situations. Despite the fact that the passive network monitoring technique is shown to perform poorly as a direct replacement for kernel instrumentation, this paper indicates the areas where passive network monitoring could be used to the great ...

13 The Zebra striped network file system

John H. Hartman, John K. Ousterhout

August 1995 **ACM Transactions on Computer Systems (TOCS)**, Volume 13 Issue 3

Publisher: ACM Press

Full text available: ▓ pdf(2.76 MB)   Additional Information: full citation, abstract, references, citings, index terms, review

Zebra is a network file system that increases throughput by striping the file data across multiple servers. Rather than striping each file separately, Zebra forms all the new data from each client into a single stream, which it then stripes using an approach similar to a log-structured file system. This provides high performance for writes of small files as well as for reads and writes of large files. Zebra also writes parity information in each stripe in the style of RAID disk arrays; this ...

Keywords: RAID, log-based striping, log-structured file system, parity computation

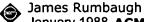**14** Using cache memory to reduce processor-memory traffic

James R. Goodman

June 1983 **ACM SIGARCH Computer Architecture News , Proceedings of the 10th annual international symposium on Computer architecture ISCA '83**, Volume 11 Issue 3

Publisher: IEEE Computer Society Press, ACM Press

Additional Information: full citation, abstract, references, citings, index terms

The importance of reducing processor-memory bandwidth is recognized in two distinct situations: single board computer systems and microprocessors of the future. Cache memory is investigated as a way to reduce the memory-processor traffic. We show that traditional caches which depend heavily on spatial locality (look-ahead) for their performance are inappropriate in these environments because they generate large bursts of bus traffic. A cache exploiting primarily temporal locality (look-behi ...

15 Controlling propagation of operations using attributes on relations

James Rumbaugh

January 1988 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications OOPSLA '88**, Volume 23 Issue 11

Publisher: ACM Press

Full text available: ▓ pdf(1.52 MB)   Additional Information: full citation, abstract, references, citings, index terms

Controlling the propagation of operations through a collection of objects connected by various relationships has been a problem both for the object-oriented and the data base communities. Operations such as copy, destroy, print, and save must propagate to some, but not all, of the objects in a collection. Such operations can be implemented using ad hoc methods on objects, at the cost of extra work and loss of clarity. The use of propagation attributes on the relationships between objects pe ...

16 Performance tradeoffs in cache design

◈ S. Prybylski, M. Horowitz, J. Hennessy

May 1988 **ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture ISCA '88,** Volume 16 Issue 2

Publisher: IEEE Computer Society Press, ACM Press

Additional Information: full citation, abstract, citings, index terms

Cache memories have become common across a wide range of computer implementations. To date, most analyses of cache performance have concentrated on time independent metrics, such as miss rate and traffic ratio. This paper presents a series of simulations that explore the interactions between various organizational decisions and program execution time. We investigate the tradeoffs between cache size and CPU/Cache cycle time, set associativity and cycle time, and between block size and main m ...

17 A simulation study of two-level caches

◈ R. T. Short, H. M. Levy

May 1988 **ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture ISCA '88,** Volume 16 Issue 2

Publisher: IEEE Computer Society Press, ACM Press

Full text available: 📄 pdf(845.70 KB)   Additional Information: full citation, abstract, references, citings, index terms

We report on a trace-driven simulation study to examine the effect of a two-level cache hierarchy in uniprocessors. A simulation model of a multiple-cycle-per-instruction processor was constructed to estimate the total cycles required to execute a synthetic benchmark. Results show that a second-level cache can be used to increase system performance when main memory access times are large relative to CPU cycle time. For example, the addition of a 4-cycle, 64K second-level cache following a 1 ...

18 BASE: using abstraction to improve fault tolerance

◈ Rodrigo Rodrigues, Miguel Castro, Barbara Liskov

October 2001 **ACM SIGOPS Operating Systems Review , Proceedings of the eighteenth ACM symposium on Operating systems principles SOSP '01,** Volume 35 Issue 5

Publisher: ACM Press

Full text available: 📄 pdf(1.47 MB)   Additional Information: full citation, abstract, references, citings, index terms

Software errors are a major cause of outages and they are increasingly exploited in malicious attacks. Byzantine fault tolerance allows replicated systems to mask some software errors but it is expensive to deploy. This paper describes a replication technique, BASE, which uses abstraction to reduce the cost of Byzantine fault tolerance and to improve its ability to mask software errors. BASE reduces cost because it enables reuse of off-the-shelf service implementations. It improves availability ...

19 Improving cache performance with balanced tag and data paths

◈ Jih-Kwon Peir, Windsor W. Hsu, Honesty Young, Shauchi Ong

September 1996 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the seventh international conference on Architectural support for programming languages and operating systems ASPLOS-VII,** Volume 31 , 30 Issue 9 , 5

Publisher: ACM Press

Full text available: 📄 pdf(1.07 MB)   Additional Information: full citation, abstract, references, citings, index terms

There are two concurrent paths in a typical cache access --- one through the data array and the other through the tag array. The path through the data array drives the selected set out of the array. The path through the tag array determines cache hit/miss and, for

set-associative caches, selects the appropriate line from within the selected set. In both direct-mapped and set-associative caches, the path through the tag array is significantly longer than that through the data array. In this paper ...

20 Classification and performance evaluation of instruction buffering techniques ■

Lizyamma Kurian, Paul T. Hulina, Lee D. Coraor, Dhamir N. Mannai

April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture ISCA '91,** Volume 19 Issue 3

Publisher: ACM Press

Full text available: .pdf(940.03 KB)   Additional Information: full citation, references, citings, index terms

Useful downloads: Adobe Acrobat   QuickTime   Windows Media Player   Real Player

P◉RTAL

USPTO

Subscribe (Full Service)   Register (Limited Service, Free)   Login

Search: ⦿ The ACM Digital Library   ○ The Guide

"attribute cache" and TLB   **SEARCH**

☜ Feedback Report a problem Satisfaction survey

Terms used **attribute cache** and **TLB**   Found **690** of **171,143**

Sort results by   [relevance ▼]   ◆ Save results to a Binder   Try an Advanced Search
                                          ☒ Search Tips   Try this search in The ACM Guide
Display results   [expanded form ▼]   ☐ Open results in a new window

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10   next
Best 200 shown   Relevance scale ☐☐☐☐☐

1 Improving cache performance with balanced tag and data paths

Jih-Kwon Peir, Windsor W. Hsu, Honesty Young, Shauchi Ong
September 1996 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review ,
Proceedings of the seventh international conference on Architectural
support for programming languages and operating systems ASPLOS-
VII**, Volume 31 , 30 Issue 9 , 5
Publisher: ACM Press

Full text available: 📄 pdf(1.07 MB)   Additional Information: full citation, abstract, references, citings, index terms

There are two concurrent paths in a typical cache access --- one through the data array
and the other through the tag array. The path through the data array drives the selected
set out of the array. The path through the tag array determines cache hit/miss and, for
set-associative caches, selects the appropriate line from within the selected set. In both
direct-mapped and set-associative caches, the path through the tag array is significantly
longer than that through the data array. In this paper ...

2 A case for two-way skewed-associative caches

André Seznec
May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th
annual international symposium on Computer architecture ISCA '93**, Volume
21 Issue 2
Publisher: ACM Press
Full text available: 📄 pdf(975.20 KB)   Additional Information: full citation, references, citings, index terms

3 Architectural support for performance tuning: a case study on the SPARCcenter 2000

A. Singhal, A. J. Goldberg
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST
annual international symposium on Computer architecture ISCA '94**, Volume
22 Issue 2
Publisher: IEEE Computer Society Press, ACM Press

Full text available: 📄 pdf(1.37 MB)   Additional Information: full citation, abstract, references, citings, index terms

Latency hiding techniques such as multilevel cache hierarchies yield high performance
when applications map well onto hierarchy implementations, but performance can suffer
drastically when they do not. Identifying and reducing mismatches between an application
and the memory hierarchy is difficult without insight into the actual behavior of the
hardware implementation. We advocate the use of hardware event counters, as a cheap,
effective and practical way to tune applications for a given hardwar ...

4 Going the distance for TLB prefetching: an application-driven study

Gokul B. Kandiraju, Anand Sivasubramaniam

May 2002 **ACM SIGARCH Computer Architecture News , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02 , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02**, Volume 30 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

The importance of the Translation Lookaside Buffer (TLB) on system performance is well known. There have been numerous prior efforts addressing TLB design issues for cutting down access times and lowering miss rates. However, it was only recently that the first exploration [26] on prefetching TLB entries ahead of their need was undertaken and a mechanism called Recency Prefetching was proposed. There is a large body of literature on prefetching for caches, and it is not clear how they can be ada ...

Keywords: application-driven study, memory hierarchy, prefetching, simulation, translation lookaside buffer

5 Characterizing the *d*-TLB behavior of SPEC CPU2000 benchmarks

Gokul B. Kandiraju, Anand Sivasubramaniam

June 2002 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS '02**, Volume 30 Issue 1

Publisher: ACM Press

Full text available: pdf(1.21 MB) Additional Information: full citation, abstract, references, citings

Despite the numerous optimization and evaluation studies that have been conducted with TLBs over the years, there is still a deficiency in an indepth understanding of TLB characteristics from an application angle. This paper presents a detailed characterization study of the TLB behavior of the SPEC CPU2000 benchmark suite. The contributions of this work are in identifying important application characteristics for TLB studies, quantifying the SPEC2000 application behavior for these characteristic ...

6 A look at several memory management units, TLB-refill mechanisms, and page table organizations

Bruce L. Jacob, Trevor N. Mudge

October 1998 **ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII**, Volume 32 , 33 Issue 5 , 11

Publisher: ACM Press

Full text available: pdf(1.90 MB) Additional Information: full citation, abstract, references, citings, index terms

Virtual memory is a staple in modem systems, though there is little agreement on how its functionality is to be implemented on either the hardware or software side of the interface. The myriad of design choices and incompatible hardware mechanisms suggests potential performance problems, especially since increasing numbers of systems (even embedded systems) are using memory management. A comparative study of the implementation choices in virtual memory should therefore aid system-level designers ...

7 Recency-based TLB preloading

Ashley Saulsbury, Fredrik Dahlgren, Per Stenström

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2

Publisher: ACM Press

Caching and other latency tolerating techniques have been quite successful in maintaining high memory system performance for general purpose processors. However, TLB misses have become a serious bottleneck as working sets are growing beyond the capacity of TLBs. This work presents one of the first attempts to hide TLB miss latency by using preloading techniques. We present results for traditional next-page TLB miss preloading - an approach shown to cut so ...

8 The TLB slice—a low-cost high-speed address translation mechanism

George Taylor, Peter Davies, Michael Farmwald
May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture ISCA '90**, Volume 18 Issue 3a
Publisher: ACM Press

The MIPS R6000 microprocessor relies on a new type of translation lookaside buffer — called a TLB slice — which is less than one-tenth the size of a conventional TLB and as fast as one multiplexer delay, yet has a high enough hit rate to be practical. The fast translation makes it possible to use a physical cache without adding a translation stage to the processor's pipeline. The small size makes it possible to include address translation on-chip, even in a tech ...

9 Increasing TLB reach using superpages backed by shadow memory

Mark Swanson, Leigh Stoller, John Carter
April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3
Publisher: IEEE Computer Society, ACM Press

The amount of memory that can be accessed without causing a TLB fault, the reach of a TLB, is failing to keep pace with the increasingly large working sets of applications. We propose to extend TLB reach via a novel Memory Controller TLB (MTLB) that lets us aggressively create superpages from non-contiguous, unaligned regions of physical memory. This flexibility increases the OS's ability to use superpages on arbitrary application data. The MTLB supports shadow pages, regions of physical address ...

10 A simulation based study of TLB performance

J. Bradley Chen, Anita Borg, Norman P. Jouppi
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture ISCA '92**, Volume 20 Issue 2
Publisher: ACM Press

This paper presents the results of a simulation-based study of various translation lookaside buffer (TLB) architectures, in the context of a modern VLSI RISC processor. The simulators used address traces, generated by instrumented versions of the SPEC marks and several other programs running on a DECstation 5000. The performance of two-level TLBs and fully-associative TLBs were investigated. The amount of memory mapped was found to be the dominant factor in TLB performance. Small first-leve ...

11 Reducing TLB and memory overhead using online superpage promotion

Theodore H. Romer, Wayne H. Ohlrich, Anna R. Karlin, Brian N. Bershad
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd**

Full text available: pdf(1.41 MB)   Additional Information: full citation, abstract, references, citings, index terms

Modern microprocessors contain small TLBs that maintain a cache of recently used translations. A TLB's *coverage* is the sum of the number of bytes mapped by each entry. Applications with working sets larger than the TLB coverage will perform poorly due to high TLB miss rates. Superpages have been proposed as a mechanism for increasing TLB coverage. A *superpage* is a virtual memory page with size and alignment that are a power of two multiple of the system's base page size. In this pap ...

12 Surpassing the TLB performance of superpages with less operating system support

Madhusudhan Talluri, Mark D. Hill

Full text available: pdf(1.50 MB)   Additional Information: full citation, abstract, references, citings, index terms

Many commercial microprocessor architectures have added translation lookaside buffer (TLB) support for superpages. Superpages differ from segments because their size must be a power of two multiple of the base page size and they must be aligned in both virtual and physical address spaces. Very large superpages (e.g., 1MB) are clearly useful for mapping special structures, such as kernel data or frame buffers. This paper considers the architectural and opera ...

13 Low power memory: Synonymous address compaction for energy reduction in data TLB

Chinnakrishnan S. Ballapuram, Hsien-Hsin S. Lee, Milos Prvulovic

Full text available: pdf(194.54 KB)   Additional Information: full citation, abstract, references, index terms

Modern processors can issue and execute multiple instructions per cycle, often performing multiple memory operations simultaneously. To reduce stalls due to resource conflicts, most processors employ multi-ported L1 caches and TLBs to enable concurrent memory accesses. In this paper, we observe that data TLB lookups within a cycle and across consecutive cycles are often *synonymous* --- they go to the same page. To exploit this finding, we propose two new mechanisms --- *intra-cycle compa* ...

Keywords: low-power TLB, multi-porting, spatial and temporal locality

14 Low power memory: An energy efficient TLB design methodology

Dongrui Fan, Zhimin Tang, Hailin Huang, Guang R. Gao

Full text available: pdf(268.72 KB)   Additional Information: full citation, abstract, references, index terms

This paper researches Translation Look-aside Buffer (TLB) of embedded processor. Based on an analysis of design-related factors: power, area, critical path and performance of our research model-Godson-I, a low-power TLB design is proposed without sacrifice of performance and timing. Using this method, the following results are achieved: power of TLB-RAM reduces 92.7% and area of TLB-RAM reduces 50%. Compared with other methods, the hit rate of this design is much higher and the accessing conflic ...

Keywords: Godson-I, TLB, embedded processor design, energy efficient, low-power consumption, single-port RAM

15 Energy efficient microarchitectural techniques: A selective filter-bank TLB system

Jung-Hoon Lee, Gi-Ho Park, Sung-Bae Park, Shin-Dug Kim

August 2003 **Proceedings of the 2003 international symposium on Low power electronics and design**

Publisher: ACM Press

Full text available: ☐ pdf(306.30 KB) Additional Information: full citation, abstract, references, citings, index terms

We present a selective filter-bank translation lookaside buffer (TLB) system with low power consumption for embedded processors. The proposed TLB is constructed as multiple banks with a small two-bank buffer, called as a filter-bank buffer, located above its associated bank. Either a filter-bank buffer or a main bank TLB can be selectively accessed based on two bits in the filter-bank buffer. Energy savings are achieved by reducing the number of entries accessed at a time, by using filtering and ...

Keywords: filtering mechanism, low power consumption, performance evaluation, translation lookaside buffer

16 Boosting superpage utilization with the shadow memory and the partial-subblock TLB

Cheol Ho Park, JaeWoong Chung, Byeong Hag Seong, YangWoo Roh, Daeyeon Park

May 2000 **Proceedings of the 14th international conference on Supercomputing**

Publisher: ACM Press

Full text available: ☐ pdf(798.29 KB) Additional Information: full citation, abstract, references, index terms

While superpage is an efficient solution to increase TLB reach, its limited flexibility for address mapping is still a hard issue. Our proposed mechanism has been developed for taking advantage of two previous approaches which resolve the issue partially: the partial-subblock TLB and the shadow memory. Through integration of them, our mechanism enjoys various benefits inherited from the both sides. By adopting Memory Controller TLB (MTLB) from the shadow memory, it allows superpages to be c ...

17 A compiler framework for restructuring data declarations to enhance cache and TLB effectiveness

David F. Bacon, Jyh-Herng Chow, Dz-ching R. Ju, Kalyan Muthukumar, Vivek Sarkar

October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available: ☐ pdf(298.15 KB) Additional Information: full citation, abstract, references, citings, index terms

It has been observed that memory access performance can be improved by restructuring data declarations, using simple transformations such as array dimension padding and inter-array padding (array alignment) to reduce the number of misses in the cache and TLB (translation lookaside buffer). These transformations can be applied to both static and dynamic array variables. In this paper, we provide a padding algorithm for selecting appropriate padding amounts, which takes into account various cache ...

18 Reducing TLB power requirements

Toni Juan, Tomas Lang, Juan J. Navarro

August 1997 **Proceedings of the 1997 international symposium on Low power electronics and design**

Publisher: ACM Press

Full text available: ☐ pdf(751.55 KB) Additional Information: full citation, references, citings

19 <u>Optimizing instruction TLB energy using software and hardware techniques</u>

I. Kadayif, A. Sivasubramaniam, M. Kandemir, G. Kandiraju, G. Chen

April 2005 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**,
Volume 10 Issue 2

Publisher: ACM Press

Full text available:📄.pdf(<u>551.77 KB</u>)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

Power consumption and power density for the Translation Look-aside Buffer (TLB) are important considerations not only in its design, but can have a consequence on cache design as well. After pointing out the importance of instruction TLB (iTLB) power optimization, this article embarks on a new philosophy for reducing the number of accesses to this structure. The overall idea is to keep a translation currently being used in a register and avoid going to the iTLB as far as possible---until there i ...

Keywords: Power consumption, cache design, compiler optimization, instruction locality, translation look-aside buffer

20 <u>Energy efficient microarchitectural techniques: Energy efficient D-TLB and data</u> <u>cache using semantic-aware multilateral partitioning</u>

Hsien-Hsin S. Lee, Chinnakrishnan S. Ballapuram

August 2003 **Proceedings of the 2003 international symposium on Low power electronics and design**

Publisher: ACM Press

Full text available:📄.pdf(<u>280.25 KB</u>)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

The memory subsystem, including address translations and cache accesses, consumes a major portion of the overall energy on a processor. In this paper, we address the memory energy issues by using a streamlined architectural partitioning technique that effectively reduces energy consumption in the memory subsystem without compromising performance. It is achieved by decoupling the d-TLB lookups and the data cache accesses, based on the semantic regions defined by programming languages and software ...

Keywords: energy optimization, low-power TLB, low-power cache, multi-ported memory structures

Results 1 - 20 of 200   Result page: **1**   <u>2</u>   <u>3</u>   <u>4</u>   <u>5</u>   <u>6</u>   <u>7</u>   <u>8</u>   <u>9</u>   <u>10</u>   <u>next</u>

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	49	(US-5752274-$ or US-6223256-$ or US-5018061-$ or US-6006312-$ or US-6041396-$ or US-6711653-$ or US-5906001-$ or US-5963984-$ or US-6189074-$ or US-6275917-$ or US-6304944-$ or US-6105113-$ or US-5897664-$ or US-5305444-$ or US-5426750-$ or US-5440707-$ or US-6205531-$ or US-6442664-$ or US-6658538-$ or US-6782453-$ or US-5617348-$ or US-5446685-$ or US-5598115-$ or US-5940873-$ or US-5974520-$ or US-6175906-$).did. or (US-6370622-$ or US-6681311-$ or US-5751996-$ or US-5918250-$ or US-5918251-$ or US-6014737-$ or US-5717885-$ or US-5946716-$ or US-6351797-$ or US-4849876-$ or US-6058437-$ or US-5479630-$ or US-5630088-$ or US-6101590-$ or US-6073226-$ or US-4899275-$ or US-5329627-$ or US-5497480-$ or US-5675763-$ or US-5696927-$ or US-6208543-$ or US-5222222-$ or US-5455834-$).did.	USPAT	OR	ON	2006/03/09 17:31
L2	20519	physical adj address	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L3	49	(US-5752274-$ or US-6223256-$ or US-5018061-$ or US-6006312-$ or US-6041396-$ or US-6711653-$ or US-5906001-$ or US-5963984-$ or US-6189074-$ or US-6275917-$ or US-6304944-$ or US-6105113-$ or US-5897664-$ or US-5305444-$ or US-5426750-$ or US-5440707-$ or US-6205531-$ or US-6442664-$ or US-6658538-$ or US-6782453-$ or US-5617348-$ or US-5446685-$ or US-5598115-$ or US-5940873-$ or US-5974520-$ or US-6175906-$).did. or (US-6370622-$ or US-6681311-$ or US-5751996-$ or US-5918250-$ or US-5918251-$ or US-6014737-$ or US-5717885-$ or US-5946716-$ or US-6351797-$ or US-4849876-$ or US-6058437-$ or US-5479630-$ or US-5630088-$ or US-6101590-$ or US-6073226-$ or US-4899275-$ or US-5329627-$ or US-5497480-$ or US-5675763-$ or US-5696927-$ or US-6208543-$ or US-5222222-$ or US-5455834-$).did.	USPAT	OR	ON	2006/03/09 17:31
L4	49	L2 and L3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L5	5622	tlb or (translation adj lookaside adj buffer)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31

L6	885	attribute near3 cache	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:33
L7	9	L2 same L6 same L5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L8	24258074	@ad<"20030703"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L9	7	L7 and L8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L10	94	attribute near5 physical adj address	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:40
L11	26	L5 and L10	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L12	759	711/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L13	851	711/206.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L14	1055	711/144.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L15	2111	711/154.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L16	657	711/3.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31

L17	4954	L12 or L13 or L14 or L15 or L16	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L18	17	L10 and L17	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L19	14	L5 and L18	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L20	7	L6 and L19	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L21	6	L8 and L20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L22	0	parallel adj attribute adj cache	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L23	0	serial adj attribute adj cache	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L24	0	basic adj attribute adj cache	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L25	326	attribute near3 lookup	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L26	3	L10 and L25	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:31
L27	1461	attribute near5 (cache or cached or caching)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:33

L28	28	10 and 27	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:33
L29	15	8 and 28	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:39
L30	153	711/207.ccls.	US-PGPUB	OR	ON	2006/03/09 17:39
L31	2	29 and 30	US-PGPUB	OR	ON	2006/03/09 17:39
L32	94	attribute near5 physical adj address$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/03/09 17:40